/* InitializeCm creates the CM thread, and then delays return to its caller, until the CM thread signals that */
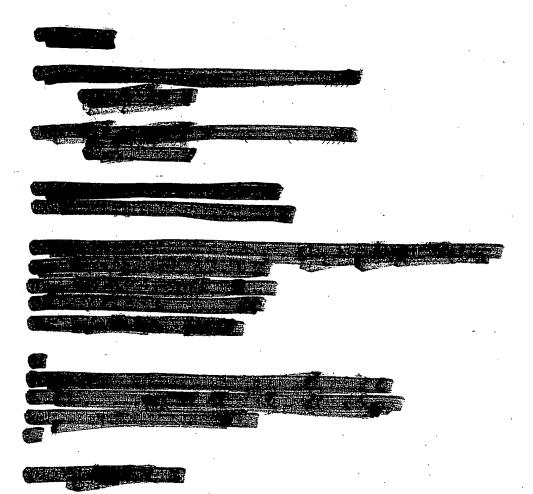/* it is ready to accept application registration with the CM.   */

int InitCm(short *NodeId, LPCLUSTERDB pClusterDb, NEDC_BOUND_SET *lpBoundSet,
          PDISKDEV pDiskList, PLOGNAME pNameList)

{

```
██████████

███████████████████████████████

     ████████████████

████████████████████████████████

████████████████████████████

████████████████████████████████████████████████
████████████████████████████
████████████████████████████
████████████████████

▇████████████████████████████████
██████████████████████████████████
██████████████████████████
█

████████████████

    if (!(status = CreateCmThread(pClub))) return status; /* Cm thread create failed */

    LogMsg("Waiting for Cm to start");

    if (!WaitForCm(pClub)) return FALSE;  // wait for Cm to initialize

███████████████████████████████████████
████████████████████████████████████████████████████
```

```
    return  TRUE; /* ready to accept application registration */

}
```